Donut

Document Understanding Transformer

Reviewed by Junsu Kim

Abstract

- Understanding document images is important but challenging task, which involves
  - Reading text
  - Understanding structure/relationship of the text
- There has been many approaches that "outsource" the reading part and then, the OCR-ed text goes through complex parsing processes to become a structured information
- These kinds of approaches are
  - Expensive in computational cost perspective
  - Inflexible in domain/language transition
  - Exposed to the error propagation from OCR (as OCR and understanding are independent)

Training

Pre-train and Fine-tune

- In pre-training stage, Donut learns *how to read the text*
  - Teacher-forcing is used, which
    - Uses ground-truth for decoder output, instead of using what predicted
- In fine-tuning, Donut learns *how to understand the whole document*
  - Three downstream tasks
    - Document classification
    - Document Information Extraction
    - Document Visual Question Answering

Architecture

- Encoder
  - Visual encoder converts input image into a set of embeddings
  - Swin Transformer is adopted, which
    - Splits input image into non-overlapping patches
    - Then, multi-head self-attention module and an MLP module are applied
    - Resulting in Z from the final encoding block which will be used as an input of decoder
- Decoder
  - Given Z, generates a token sequence
  - BART is adopted

Further Studies

Considered helpful to my project!

- Donut outperforms (speed and accuracy) state-of-art OCR-based engines (BERT, LayoutLMv2)
- The complexity of my project might be located between CORD and Ticket. Characteristics of CORD dataset:
  - Consists of 0.8K train, 0.1K valid, and 0.1K test; has 30 fields (menu name, count, total, price…); has complex structure (nested structure: under each item, it can has name, count, price, and so on.)
- My project(certificate) has 5 field items to be extracted as of now. However, format of certificate vary a lot according to the publishing body

Further Studies-2

Considered helpful to my project!

- Donut shows performance growth with larger input size (resolution)
  - Maximum at 1280 x 960; comparable but less when it as 2560 x 1920
- It gives robust accuracy in low resourced situation
  - It seems that 160 samples of (1280 or 2560 resolution) was achieved accuracy over 80%
  - So I need to check the quality of my input and increase the resolution or gather high quality input
    - Do I need to utilize super-resolution method for my inputs?
- Refer to these to delve into efficient structured information from OCR-ed text
  - "Cost-effective end-to-end information extraction for semi-structured document images"
  - "Spatial dependency parsing for semi-structured document information extraction"

Appendix

- Code Reference
    - https://colab.research.google.com/drive/16iPnVD680MnCqxHcLaq9qn9zkRaIGeab?usp=sharing#scrollTo=h072rV0FMNYb